

IMAGE-PROCESSING METHOD AND IMAGE PROCESSOR

BACKGROUND OF THE INVENTION

[0001] 1. Field of the Invention

[0002] The present invention relates to an image-processing method and image processor for filtering image data to operatively remove encoding-caused noises.

[0003] 2. Description of the Related Art

[0004] Encoded and decoded images over a videophone and a TV conference have heretofore been used to provide a reduced amount of image information in order to achieve efficient image transmission.

[0005] A majority of such an image encoding-decoding method in recent years uses discrete cosine transform.

[0006] The discrete cosine transform achieves a substantial reduction in image information amount. At the same time, the discrete cosine transform brings up a problem of peculiar Gaussian noise called a mosquito noise. The noise appears on a decoded image, and spoils subjective image quality.

[0007] The Gaussian noise as represented by the mosquito noise is removable using a spatial filter. The spatial filter functions to remove high-frequency components.

[0008] The following specifically discusses filtering, in which image data having a QCIF (quarter common intermediate format) size (144 lines by 176 pixels) is filtered as an illustration. Each gang of three-pixels-by-three-pixels image data is filtered. A pixel located at the center of three pixels-by-three pixels has target image data subject to filtering.

[0009] Fig. 25 is a block diagram illustrating a prior art image processor designed to filter target image data subject to filtering.

[0010] As shown in Fig. 25, the prior art image processor includes a processor 500, an image memory 501, line buffers 502 and 504, and a filtering circuit 503.

[0011] The processor 500 includes a source memory 505 and a destination memory 506.

[0012] The processor 500 reads out encoded image data from the image memory 501 and writes the same encoded image data into the source memory 505.

[0013] The processor 500 decodes the encoded image data in the source memory 505.

[0014] The processor 500 at first stores the decoded image data into the destination memory 506, and then transfers the same image data to the image memory 501.

[0015] The decoded image data in the image memory 501 are filtered.

[0016] In order to filter the decoded image data, 3-lines-by-176-pixels image data are transmitted to the line buffer 502 from the image memory 501.

[0017] The line buffer 502 stores the transmitted 3-lines-by-176-pixels image data.

[0018] The filtering circuit 503 reads out the three pixels-by-three pixels image data from the line buffer 502 to filter a piece of target image data subject to filtering.

[0019] The filtering circuit 503 repeats the above steps, thereby filtering 174-pieces of target image data subject to filtering that are stored in the line buffer 502.

[0020] Image data located in a central line of 3-lines-by-176-pixels (except for image data at opposite ends of the central line) among the 3-lines-by-176-pixels image data in the line buffer 502 are target image data subject to filtering.

[0021] The filtering circuit 503 feeds the filtered 174-pieces of image data into the line buffer 504.

[0022] At the same time, the filtering circuit 503 feeds the image data at opposite ends of the central line as previously discussed into the line buffer 504 without filtering them.

[0023] As a result, the line buffer 504 stores a line (176-pieces) of filtered image data.

[0024] As discussed above, the past practice is to transmit the image data to the line buffer 502 for each gang of 3-lines-by-176-pixels, thereby filtering the QCIF-sized image data.

[0025] Details of the filtering are now discussed.

[0026] Fig. 26 is a block diagram illustrating the filtering circuit 503 as shown in Fig. 25. Fig. 26 illustrates the same reference characters on the same components as those shown in Fig. 25.

[0027] As shown in Fig. 26, the filtering circuit 503 includes a data readout unit 507, a filtering unit 508, and a data write unit 509.

[0028] As shown in Fig. 26, assume that the line buffer 502 is designed to store the 3-lines-by-176-pixels image data at maximum.

[0029] The data readout unit 507 reads out image data from the line buffer 502.

[0030] The filtering unit 508 filters the read image data.

[0031] The data write unit 509 writes the filtered image data to the line buffer 504.

[0032] The line buffer 504 is constructed to store a line (176-pixels) of image data at maximum.

[0033] The prior art image processor repeats the following steps: the data readout unit 507 reads out three-pixels-by-three-pixels image data from the line buffer 502; the filtering unit 508 filters the read three pixels-by-three pixels image data; and the data write unit 509 writes a pixel to the line buffer 504.

[0034] The following discusses the filtering with reference to a specific example.

[0035] Fig. 27 is a descriptive illustration showing how image data in "N"-line of a QCIF image are filtered in a conventional manner.

[0036] Fig. 28 is a descriptive illustration showing how image data in "N+1"-line of the QCIF image are filtered in a conventional manner.

[0037] As shown in Fig. 27, the data readout unit 507 reads out three-pixels-by-three-pixels image data 600 to perform the first filtering.

[0038] The filtering unit 508 filters image data (N, 2) using surrounding image data (N- 1, 1), (N, 1) (N+1, 1), (N-1, 2), (N+1, 2), (N-1, 3), (N, 3) and (N+1, 3).

[0039] The data write unit 509 writes the filtered image data (FN, 2) to the line buffer

504.

[0040] The data readout unit 507 reads out three-pixels-by-three-pixels image data 601 to execute the second filtering.

[0041] The filtering unit 508 filters image data (N, 3) using surrounding image data (N- 1, 2), (N, 2) (N+1, 2), (N-1, 3), (N+1, 3), (N-1, 4), (N, 4) and (N+1, 4).

[0042] The data write unit 509 writes the filtered image data (FN, 3) to the line buffer 504.

[0043] The above steps are repeated 174-times. As a result, the line buffer 504 stores "N"-line of image data (N, 1), (FN, 2) to (FN, 175), and (N, 176).

[0044] Upon completion of the filtering of the image data in "N"-line, image data in "N+1"-line are filtered, as shown in Fig. 28.

[0045] Deleting "N-1"-line, "N+2"-line is added as the last line.

[0046] In this way, lines to be filtered are selected in sequence to repeat the above steps.

[0047] In general, pixels at both ends of the central line where surrounding pixels are absent are not filtered. In addition, pixels in the first line as well as the 144th line are not filtered.

[0048] Accordingly, 142-lines-by-174-pixels image data are filtered. This means that separate steps must be taken to decode the QCIF image.

[0049] As previously discussed, filtering the image data using the prior art image processor requires one line buffer (the line buffer 502 according to the above example) adapted for plural lines (three lines according to the above example) and another line buffer (the line buffer 504 according to the above example) designed to store a line of filtered image data, in accordance with the number of pixels to be filtered at a time (three pixels-by-three pixels according to the above example). For details, refer to the published Japanese Patent Application Laid-Open No. 2000-251065.

[0050] Such a requirement causes a problem of an area increase in an image processor.

[0051] Another problem is that the image data are filtered in a lengthy time because each gang of three-pixels-by-three-pixels image data is read out, for each time of the filtering, from the line buffer 502 to repeat the above steps. For details, refer to the published Japanese Patent Application Laid-Open No. 7-111586.

[0052] Furthermore, in order to process edgewise pixels, only a predetermined number of pixels are processed.

[0053] This causes a further problem that a change in image size precludes the filtering of the image data.

OBJECTS AND SUMMARY OF THE INVENTION

[0054] In view of the above, an object of the present invention is to provide an image processor and image-processing method for inhibiting an area increase in an image processor, processing image data in a reduced time, and coping with a change in image size.

[0055] A first aspect of the present invention provides an image-processing method comprising: transferring image data to a storage unit, the image data being formed by several pieces of image data, the image data being separated from several lines of image data; sequentially reading out the several pieces of image data from the storage unit; and filtering target image data subject to filtering using predetermined pieces of the read image data.

[0056] The above method allows several lines of image data to be divided into partitions in accordance with the storage capacity of a destination or rather the storage unit in order to transfer the partitions to the storage unit. This feature eliminates a storage unit having large storage capacity sufficient to store the several lines of image data. Consequently, an area increase in an image processor is inhibited.

[0057] The above method allows an existing storage unit to be used as a destination. This feature eliminates a dedicated storage unit designed for filtering, which serves as a destination. Consequently, an area increase in an image processor is again inhibited.

[0058] The above method sequentially reads out several pieces of image data from the storage unit. This feature filters several pieces of target image data subject to filtering without reading out the same image data several times. As a result, the target image data subject to filtering can be filtered in a reduced time.

[0059] The above method makes it feasible to filter the target image data subject to filtering without fixing the number of pixels in a line. This feature filters the target image data subject to filtering independently of an image size.

[0060] A second aspect of the present invention is to provide an image-processing method comprising: sequentially entering several pieces of image data that are used to filter several pieces of target image data subject to filtering; shifting the sequentially entered several pieces of image data; parallel-feeding predetermined pieces of the shifted image data; and filtering the target image data subject to filtering using the parallel-fed predetermined pieces of the shifted image data.

[0061] The above method sequentially reads out several pieces of image data from the storage unit for use in filtering several pieces of target image data subject to filtering. This feature filters the several pieces of target image data subject to filtering without reading out the same image data several times. As a result, the target image data subject to filtering can be filtered in a shorter time.

[0062] A third aspect of the present invention is to provide an image processor comprising: a first storage unit operable to store image data formed by several pieces of image data, the image data being separated from several lines of image data; a data readout unit operable to sequentially read out the several pieces of image data stored by the first storage unit; a filtering unit operable to filter target image data subject to filtering using predetermined pieces of the image data read out from the data readout unit; a second storage unit operable to store the filtered target image data subject to filtering; and a data write unit operable to write the filtered target image data subject to filtering to the second storage unit.

[0063] The above image processor allows several lines of image data to be divided into partitions in accordance with the storage capacity of a destination or rather the first storage unit in order to transfer the partitions to the first storage unit. This feature eliminates a first storage unit having large storage capacity sufficient to store the several lines of image data.

[0064] The above image processor allows an existing storage unit to be used as a destination or rather the first storage unit. This feature eliminates a dedicated first storage unit designed for filtering, which serves as a destination.

[0065] The above image processor filters several pieces of target image data subject to filtering, which are included in image data separated from several lines of image data. This feature eliminates the need to provide a second storage unit having large storage capacity sufficient to store a line of image data.

[0066] The above image processor allows an existing storage unit to be used as the second storage unit where the image data are written. This feature eliminates the need to provide a dedicated second storage unit designed for filtering, to which the image data are written.

[0067] Consequently, the above features inhibit an area increase in an image processor.

[0068] The above image processor sequentially reads out several pieces of image data from the first storage unit. This feature filters several pieces of target image data subject to filtering without reading out the same image data several times. As a result, the target image data subject to filtering can be filtered in a reduced time.

[0069] The above image processor makes it feasible to filter the target image data subject to filtering without providing a fixed number of pixels in a line. This feature filters the target image data subject to filtering independently of an image size.

[0070] A fourth aspect of the present invention is to provide an image processor as defined in the third aspect of the present invention, wherein the filtering unit comprises a data output control unit operable to feed filtered target image data subject to filtering,

but operable not to feed a predetermined piece of non-filtered image data in accordance with mode information.

[0071] According to the above image processor, the data output control unit removes unnecessary image data produced in the course of filtering the target image data subject to filtering. As a result, only the required image data are provided. This feature produces line data in a reduced time.

[0072] A fifth aspect of the present invention is to provide an image processor as defined in the third aspect of the present invention, wherein the filtering unit comprises a data output control unit operable to practice a countdown from an initial value each time when the image data is fed, the initial value being determined in accordance with the number of pixels to be processed, the data output control unit being operable to feed only a predetermined piece of non-filtered image data as well as filtered target image data subject to filtering in accordance with results from the countdown.

[0073] According to the above image processor, the data output control unit removes unnecessary image data produced in the course of filtering the target image data subject to filtering. As a result, only the required image data are provided. This feature produces line data in a reduced time.

[0074] The above image processor initially sets up the number of pixels to be processed. Such a simple setup provides subsequent edgewise pixel processing. This feature allows edgewise pixel-processing information to be set up a smaller number of times.

[0075] A sixth aspect of the present invention is to provide an image processor as defined in the third aspect of the present invention, wherein the filtering unit comprises a image data-retaining unit operable to retain several pieces of image data for use at the time of the next filtering, the several pieces of image data for use at the time of the next filtering being selected from among several pieces of image data read out by the data readout unit from the first storage unit.

[0076] At the end of the present filtering, the above image processor retains image data required for the following filtering, thereby using the retained image data at the time of the following filtering. This feature allows the data to be transferred a smaller number of times.

[0077] Embodiments of the present invention exemplify a QCIF (quarter common intermediate format) size in which image data to be filtered consists of 144-lines-by-176-pixels.

[0078] In the 144-lines-by-176 pixels QCIF size, linearly aligned 176-pixels forms a line; and a plurality of the lines are arranged in parallel to form 144-lines, thereby providing an image plane.

[0079] The above, and other objects, features and advantages of the present invention will become apparent from the following description read in conjunction with the accompanying drawings, in which like reference numerals designate the same elements.

BRIEF DESCRIPTION OF THE DRAWINGS

[0080] Fig. 1 is a block diagram illustrating an image processor according to a first embodiment of the present invention;

[0081] Fig. 2 is a schematic illustration showing how the image processor according to the first embodiment filters image data;

[0082] Fig. 3(a) is an illustration showing a flow of the first filtering provided by the image processor according to the first embodiment;

[0083] Fig. 3(b) is an illustration showing a flow of the second filtering provided by the image processor according to the first embodiment;

[0084] Fig. 3(c) is an illustration showing a flow of the third filtering provided by the image processor according to the first embodiment;

[0085] Fig. 4 is a block diagram illustrating a filtering circuit of the image processor according to the first embodiment;

[0086] Figs. 5(a), 5(b), and 5(c) are descriptive illustration showing the first filtering provided by the image processor according to the first embodiment;

[0087] Figs. 6(a), 6(b), and 6(c) are descriptive illustration showing the second filtering provided by the image processor according to the first embodiment;

[0088] Figs. 7(a), 7(b), and 7(c) are descriptive illustration showing the third filtering provided by the image processor according to the third embodiment;

[0089] Fig. 8(a) is an illustration showing a flow of the first filtering provided by an image processor according to a second embodiment;

[0090] Fig. 8(b) is an illustration showing a flow of the second filtering provided by the image processor according to the second embodiment;

[0091] Fig. 8(c) is an illustration showing a flow of the third filtering provided by the image processor according to the second embodiment;

[0092] Fig. 9 is a block diagram illustrating a filtering circuit of the image processor according to the second embodiment;

[0093] Figs. 10(a), 10(b), 10(c), 10(d), and 10(e) are descriptive illustrations showing the first filtering provided by the image processor according to the second embodiment;

[0094] Figs. 11(a), 11(b), 11(c), 11(d), and 11(e) are descriptive illustrations showing the first filtering provided by the image processor according to the second embodiment;

[0095] Figs. 12(a), 12(b), 12(c), 12(d), and 12(e) are descriptive illustrations showing the third filtering provided by the image processor according to the second embodiment;

[0096] Fig. 13(a) is a descriptive illustration showing mode information 2'b00 in the image processor according to the second embodiment;

[0097] Fig. 13(b) is a descriptive illustration showing mode information 2'b01 in the image processor according to the second embodiment;

[0098] Fig. 13(c) is a descriptive illustration showing mode information 2'b10 in the image processor according to the second embodiment;

[0099] Fig. 13(d) is a descriptive illustration showing mode information 2'b11 in the image processor according to the second embodiment;

[0100] Fig. 14(a) is a descriptive illustration showing the usage of mode information 2'b01 in the image processor according to the second embodiment;

[0101] Fig. 14(b) is a descriptive illustration showing the usage of mode information 2'b11 in the image processor according to the second embodiment;

[0102] Fig. 14(c) is a descriptive illustration showing the usage of mode information 2'b10 in the image processor according to the second embodiment;

[0103] Fig. 15 is a block diagram illustrating a filtering circuit of an image processor according to a third embodiment;

[0104] Fig. 16 is a schematic illustration showing how the image processor according to the third embodiment filters image data;

[0105] Fig. 17(a) is an illustration showing a flow of the first filtering provided by the image processor according to the third embodiment;

[0106] Fig. 17(b) is an illustration showing a flow of the second filtering provided by the image processor according to the third embodiment;

[0107] Fig. 17(c) is an illustration showing a flow of the third filtering provided by the image processor according to the third embodiment;

[0108] Figs. 18(a), 18(b), and 18(c) are descriptive illustrations showing characteristics of filtering steps taken by the image processor according to the third embodiment;

[0109] Fig. 19 is a block diagram illustrating a filtering circuit of an image processor according to a fourth embodiment;

[0110] Fig. 20 is a schematic illustration showing how the image processor according to the fourth embodiment filters image data;

[0111] Fig. 21(a) is an illustration showing a flow of the first filtering provided by the

image processor according to the fourth embodiment;

[0112] Fig. 21(b) is an illustration showing a flow of the second filtering provided by the image processor according to the fourth embodiment;

[0113] Fig. 21(c) is an illustration showing a flow of the third filtering provided by the image processor according to the fourth embodiment;

[0114] Figs. 22(a), 22(b), 22(c), 22(d), and 22(e) are descriptive illustrations showing the first filtering provided by the image processor according to the fourth embodiment;

[0115] Figs. 23(a), 23(b), 23(c), 23(d), 23(e), and 23(f) are descriptive illustrations showing the second filtering provided by the image processor according to the fourth embodiment;

[0116] Figs. 24(a), 24(b), 24(c), 24(d), 24(e), and 24(f) are descriptive illustrations showing the third filtering provided by the image processor according to the fourth embodiment;

[0117] Fig. 25 is a block diagram illustrating a prior art image processor;

[0118] Fig. 26 is a block diagram illustrating a filtering circuit of the prior art image processor;

[0119] Fig. 27 is a descriptive illustration showing how the prior art image processor filters "N"-line of target pixels subject to filtering; and,

[0120] Fig. 28 is a descriptive illustration showing how the prior art image processor filters "N+1"-line of target pixels subject to filtering.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

[0121] Embodiments of the present invention are discussed in accordance with the accompanying drawings.

[0122] (First embodiment)

[0123] Fig. 1 is a block diagram illustrating an image processor according to a first embodiment.

[0124] As illustrated in Fig. 1, the image processor includes a processor 1, an image

memory 2, and a filtering circuit 3.

[0125] The processor 1 includes a source memory 10, a destination memory 11, and a control unit 12.

[0126] Next, the entire behavior of the image processor is briefly discussed.

[0127] The processor 1 reads out encoded image data from the image memory 2 and writes the same encoded image data into the source memory 10.

[0128] The processor 1 decodes the encoded image data in the source memory 10.

[0129] The processor 1 at first places the decoded image data into the destination memory 11, and then transfers the same image data to the image memory 2 from the destination memory 11.

[0130] The decoded image data transferred to the image memory 2 is filtered.

[0131] In order to filter the decoded image data, 3-lines-by-176-pixels image data are divided into partitions in accordance with the storage capacity of the source memory 10. The divided image data are transmitted to the source memory 10.

[0132] The control unit 12 reads out the divided image data from the source memory 10, and then transmits the same image data to the filtering circuit 3.

[0133] The filtering circuit 3 filters the divided image data to remove noises.[0133]

[0134] The control unit 12 writes the divided image data free of the noises to the destination memory 11, and further to the image memory 2.

[0135] The above steps are repeated to remove the noises from QCIF-sized image data.

[0136] An image output circuit (not shown) feeds the image data free of the noises into a display device (not shown) from the image memory 2. As a result, the image data free of the noises are displayed on the display device (not shown). The display device is, e.g., an LCD (liquid crystal display).

[0137] Next, the filtering is briefly discussed.

[0138] Fig. 2 is a schematic diagram illustrating how the image processor according to the present embodiment filters the image data.

[0139] Assume that the image memory 2 of Fig. 1 stores the decoded, QCIF-sized image data.

[0140] For convenience of description, Fig. 2 illustrates 3-lines-by-176-pixels image data 100 among the QCIF-sized image data stored in the image memory 2.

[0141] In Fig. 2, each piece of image data is illustrated in circle. A circle filled with slant lines denotes image data to be filtered (hereinafter called "target image data subject to filtering"). A blacked out circle denotes filtered target image data subject to filtering (hereinafter simply called "filtered image data").

[0142] In Fig. 2, the source memory 10 of Fig. 1 has a storage capacity sufficient to store three-pixels-by-six-pixels image data at maximum.

[0143] Each gang of three-pixels-by-three-pixels image data is filtered. A pixel located at the center of three pixels-by-three pixels has target image data subject to filtering.

[0144] As illustrated in Fig. 2, three-pixels-by-five-pixels image data 101 divided from the 3-lines-by-176-pixels image data 100 is at first transferred to the source memory 10 from the image memory 2.

[0145] The image data 101 is sent, for each piece of image data, from the source memory 10 to the filtering circuit 3.

[0146] The filtering circuit 3 filters a piece of target image data subject to filtering using the given three-pixels-by-three-pixels image data.

[0147] In this way, all three pieces of target image data subject to filtering in the image data 101 are filtered.

[0148] Three-pixels-by-six-pixels image data 102 divided from the 3-lines-by-176-pixels image data 100 is transferred to the source memory 10 from the image memory 2.

[0149] The image data 102 is sent, for each piece of image data, from the source memory 10 to the filtering circuit 3.

[0150] The filtering circuit 3 filters a piece of target image data subject to filtering

using the given three-pixels-by-three-pixels image data.

[0151] In this way, all four pieces of target image data subject to filtering in the image data 102 are filtered.

[0152] The above steps are repeated to filter all target image data subject to filtering in the 3-lines-by-176-pixels image data 100.

[0153] As a result, a line of filtered image data 105 is obtained.

[0154] In the image data 100, image data in a second line at the both ends thereof are not filtered.

[0155] Details of the filtering are now discussed with reference to Figs. 1, 3, 4, 5, 6, and 7, in which the source memory 10 is designed to store 3-lines-by-64-pixels image data at maximum.

[0156] Each gang of three-pixels-by-three-pixels image data is filtered. A pixel located at the center of three pixels-by-three pixels has target image data subject to filtering.

[0157] Figs. 3(a), 3(b), and 3(c) are illustrations showing a flow of filtering steps. Fig. 3(a) is an illustration showing a flow of first filtering. Fig. 3(b) is an illustration showing a flow of second filtering. Fig. 3(c) is an illustration showing a flow of third filtering. Fig. 3 illustrates the same reference characters on the same components as those described in Fig. 1.

[0158] Fig. 4 is a block diagram illustrating the filtering circuit 3 of Fig. 1. As illustrated in Fig. 4, the filtering circuit 3 includes a shift register 30 and a filter calculator 31. The shift register 30 includes nine flip-flops FF1 to FF9.

[0159] The filter calculator 31 sometimes feeds image data designated by sign "#". The filter calculator 31 feeds unfiltered image data as well as filtered image data. The unfiltered image data is discussed later.

[0160] Fig. 5(a), 5(b), and 5(c) are time charts showing the first filtering. Fig. 5(a) is an illustration showing, for each cycle, image data sent to the filter calculator 31 of Fig. 4. Fig. 5(b) is an illustration showing filtered image data fed from the filter calculator 31.

Fig. 5(c) is an illustrating showing an effective data-indicating signal fed from the filter calculator 31.

[0161] Fig. 6(a), 6(b), and 6(c) are time charts showing the second filtering. Fig. 6(a) is an illustration showing, for each cycle, image data sent to the filter calculator 31. Fig. 6(b) is an illustration showing filtered image data fed from the filter calculator 31. Fig. 6(c) is an illustrating showing an effective data-indicating signal fed from the filter calculator 31.

[0162] Fig. 7(a), 7(b), and 7(c) are time charts showing the third filtering. Fig. 7(a) is an illustration showing, for each cycle, image data sent to the filter calculator 31. Fig. 7(b) is an illustration showing filtered image data fed from the filter calculator 31. Fig. 7(c) is an illustrating showing an effective data-indicating signal fed from the filter calculator 31.

[0163] The following discusses details of the first filtering with reference to Fig. 3(a), Fig. 4, and Fig. 5. The first filtering refers to filtering using image data that covers from the first pixel to the sixty-fourth pixel. These pixels are separated from the 3-lines-by-176-pixels image data among the QCIF-sized image data.

[0164] As illustrated in Fig. 3(a), the image memory 2 of Fig. 1 transmits 3-pixels-by-64-pixels image data to the source memory 10, in which the transmitted 3-pixels-by-64-pixels image data are stored.

[0165] The stored image data is obtained by dividing the 3-lines-by-176-pixels image data among the QCIF-sized image data in the image memory 2 in accordance with the storage capacity of the source memory 10.

[0166] The stored image data consists of 192-pieces of image data: (N-1, 1) to (N-1, 64); (N, 1) to (N, 64); and (N+1, 1) to (N+1, 64).

[0167] The above image data includes target image data subject to filtering as illustrated by (N, 2) to (N, 63).

[0168] For convenience of description, the image data from (N-1, 1) to (N+1, 64) in

the source memory 10 are sometimes designated as image data "D".

[0169] The data readout unit 120 reads out 3-pixels-by-64-pixels image data from the source memory 10. More specifically, the image data are read out up to (N-1, 64), (N, 64), and (N+1, 64) in the order of (N-1, 1), (N, 1), (N+1, 1), (N-1, 2), (N, 2), (N+1, 2), (N-1, 3), (N, 3), (N+1, 3), (N-1, 4), (N, 4), (N+1, 4), and so on.

[0170] The data readout unit 120 feeds the read image data "D" in sequence into the filtering circuit 3.

[0171] Pursuant to the present embodiment, the control unit 12 of Fig. 1 functions as the data readout unit 120.

[0172] The filtering circuit 3 in receipt of filtering start signal "Ss" from the data readout unit 120 sequentially receives image data "D" from the source memory 10. Filtering start signal "Ss" directs the filtering circuit 3 to start the filtering.

[0173] As illustrated in Fig. 4, the shift register 30 sequentially receives image data "D". The shift register 30 provides a shift action in accordance with clock "CLK".

[0174] Flip-flops "FF1" to "FF9" feed the retained image data "D1" to "D9" in parallel into the filter calculator 31.

[0175] The filter calculator 31 filters the target image data subject to filtering using the nine pieces of image data "D1" to "D9", thereby providing a piece of filtered image data D# as a result of the filtering.

[0176] Image data "D" from the source memory 10 sequentially enters the shift register 30. As a result, all of the target image data subject to filtering included in the image data in the source memory 10 are filtered, thereby providing filtered image data "D#".

[0177] The data write unit 121 of Fig. 3(a) sequentially writes filtered image data D# to the destination memory 11 of Fig. 1 in accordance with effective data-indicating signal "SE" from the filter calculator 31.

[0178] According to the present embodiment, the control unit 12 of Fig. 1 functions as the data write unit 121.

[0179] The following discusses further specific steps taken by the filtering circuit 3.

[0180] As illustrated in Fig. 5(a), at cycle "CY1", the shift register 30 feeds nine pieces of serially aligned image data (N-1, 1) to (N+1, 3) into the filter calculator 31 in parallel.

[0181] The filter calculator 31 filters target image data subject to filtering (N, 2) using the nine pieces of image data (N-1, 1) to (N+1, 3), thereby providing filtered image data (FN, 2) as shown in Fig. 5(b).

[0182] As illustrated in Fig. 5(c), the data write unit 121 writes filtered image data (FN, 2) to the destination memory 11 in accordance with effective data-indicating signal "SE" from the filter calculator 31.

[0183] The data write unit 121 performs the writing when effective data-indicating signal "SE" is at the level of "H (high)".

[0184] As shown in Fig. 5(a), at the next cycle "CY2", the shift register 30 feeds nine pieces of image data (N, 1) to (N+1, 4) in parallel.

[0185] At the following cycle "CY3", the shift register 30 feeds nine pieces of image data (N+1, 1) to (N+2, 4) in parallel.

[0186] At cycle "CY4", the shift register 30 feeds nine pieces of image data (N+2, 1) to (N+3, 4) in parallel.

[0187] The filter calculator 31 filters target image data subject to filtering (N, 3) using nine pieces of image data (N+1, 1) to (N+3, 4), thereby providing filtered image data (FN, 3) as shown in Fig. 5(b).

[0188] As illustrated Fig. 5(c), the data write unit 121 writes filtered image data (FN, 3) to the destination memory 11 in accordance with effective data-indicating signal "SE" from the filter calculator 31.

[0189] The above steps are repeated until filtering end signal "Se" enters the filtering circuit 3 from the data readout unit 120.

[0190] More specifically, as shown in Fig. 5(a), the filtering circuit 3 repeats the

filtering sixty-two times (where each time consists of three cycles), thereby providing sixty-two pieces of filtered image data (FN, 2) to (FN, 63). As a result, the destination memory 11 stores filtered image data (FN, 2) to (FN, 63).

[0191] As illustrated in Fig. 5(b), the filter calculator 31 feeds and non-filters two pieces of image data (N, 1) and (N, 64) among the image data in the source memory 10 of Fig. 3(a). Image data (N, 1) and (N, 64) are located in the central line of the image data in the source memory 10 at both ends of the central line.

[0192] The data write unit 121 writes the non-filtered image data (N, 1), (N, 64) to the destination memory 11 in accordance with effective data-indication signal "SE" as shown in Fig. 5(c).

[0193] As a result, as illustrated Fig. 3(a), the destination memory 11 stores the filtered image data (FN, 2) to (FN, 63) and the non-filtered image data (N, 1), (N, 64).

[0194] The image data consisting of (FN, 2) to (FN, 63), (N, 1), and (N, 64) as just discussed are transmitted to the image memory 2.

[0195] The following discusses details of the second filtering with reference to Figs. 3(b), Fig. 4, and Fig. 6. The second filtering refers to filtering using image data that covers from the sixty-third pixel to the one hundred and twenty-sixth pixel. These pixels are separated from the 3-lines-by-176-pixels image data among the QCIF-sized image data.

[0196] As illustrated in Fig. 3(b), the image memory 2 of Fig. 1 transmits 3-pixels-by-64-pixels image data to the source memory 10, in which the transmitted 3-pixels-by-64-pixels image data is stored.

[0197] The stored image data is obtained by dividing the 3-lines-by-176-pixels image data among the QCIF-sized image data in the image memory 2 in accordance with the storage capacity of the source memory 10.

[0198] The stored image data consists of 192-pieces of image data: (N-1, 63) to (N-1, 126); (N, 63) to (N, 126); (N+1, 63) to (N+1, 126).

[0199] The above image data includes target image data subject to filtering (N, 64) to (N, 125).

[0200] For convenience of description, the image data of (N-1, 63) to (N+1, 126) in the source memory 10 are sometimes designated as image data "D".

[0201] The data readout unit 120 reads out 3-pixels-by-64-pixels image data from the source memory 10. More specifically, the image data are read out up to (N-1, 126), (N, 126), and (N+1, 126) in the order of (N-1, 63), (N, 63), (N+1, 63), (N-1, 64), (N, 64), (N+1, 64), (N-1, 65), (N, 65), (N+1, 65), (N-1, 66), (N, 66), (N+1, 66), and so on.

[0202] The data readout unit 120 feeds the read image data (N-1, 63) etc. in sequence into the filtering circuit 3.

[0203] As illustrated in Fig. 6(a), at cycle "CY1", the shift register 30 feeds nine pieces of serially aligned image data (N-1, 63) to (N+1, 65) into the filter calculator 31 in parallel.

[0204] The filter calculator 31 filters target image data subject to filtering (N, 64) using the nine pieces of image data (N-1, 63) to (N+1, 65), thereby providing filtered image data (FN, 64) as shown in Fig. 6(b).

[0205] As illustrated in Fig. 6(c), the data write unit 121 writes the filtered image data (FN, 64) to the destination memory 11 in accordance with effective data-indicating signal "SE" from the filter calculator 31.

[0206] As shown in Fig. 6(a), at the next cycle "CY2", the shift register 30 feeds nine pieces of image data (N, 63) to (N+1, 66) in parallel.

[0207] At the following cycle "CY3", the shift register 30 feeds nine pieces of image data (N+1, 63) to (N, 66) in parallel.

[0208] At cycle "CY4", the shift register 30 feeds nine pieces of image data (N-1, 64) to (N+1, 66) in parallel.

[0209] The filter calculator 31 filters target image data subject to filtering (N, 65) using the nine pieces of image data (N-1, 64) to (N+1, 66), thereby providing filtered image

data (FN, 65) as shown in Fig. 6(b).

[0210] As illustrated Fig. 6(c), the data write unit 121 writes filtered image data (FN, 65) to the destination memory 11 in accordance with effective data-indicating signal "SE" from the filter calculator 31.

[0211] The above steps are repeated until filtering end signal "Se" enters the filtering circuit 3 from the data readout unit 120.

[0212] More specifically, as shown in Fig. 6(a), the filtering circuit 3 repeats the filtering sixty-two times (where each time consists of three cycles), thereby providing sixty-two pieces of filtered image data (FN, 64) to (FN, 125). As a result, the destination memory 11 stores filtered image data (FN, 64) to (FN, 125).

[0213] As shown in Fig. 6(b), the filter calculator 31 feeds and non-filters two pieces of image data (N, 63) and (N, 126) among the image data in the source memory 10 of Fig. 3(a). Image data (N, 63), (N, 126) are located in the central line of the image data in the source memory 10 at both ends of the central line.

[0214] The data write unit 121 writes the non-filtered image data (N, 63), (N, 126) to the destination memory 11 in accordance with effective data-indication signal "SE" as shown in Fig. 6(c).

[0215] As a result, as illustrated Fig. 3(b), the destination memory 11 stores the filtered image data (FN, 64) to (FN, 125) and the non-filtered image data (N, 63), (N, 126).

[0216] The image data of (FN, 64) to (FN, 125), (N, 63), and (N, 126) as discussed above are transmitted to the image memory 2.

[0217] The following discusses details of the third filtering with reference to Fig. 3(c), Fig. 4, and Fig. 7. The third filtering refer to filtering using image data that covers from the one hundred and twenty-fifth pixel to the one hundred and seventy-sixth pixel. These pixels are separated from the 3-lines-by-176-pixels image data among the QCIF-sized image data.

[0218] As illustrated in Fig. 3(c), the image memory 2 of Fig. 1 transmits

3-pixels-by-52-pixels image data to the source memory 10, in which the transmitted 3-pixels-by-52-pixels image data are stored.

[0219] The stored image data is obtained by dividing the 3-lines-by-176-pixels image data among the QCIF-sized image data in the image memory 2 in accordance with the storage capacity of the source memory 10.

[0220] The stored image data consists of 156-pieces of image data: (N-1, 125) to (N-1, 176); (N, 125) to (N, 176); (N+1, 125) to (N+1, 176).

[0221] The above image data includes target image data subject to filtering (N, 126) to (N, 175).

[0222] For convenience of description, the image data of (N-1, 125) to (N+1, 176) in the source memory 10 are sometimes designated as image data "D".

[0223] The data readout unit 120 reads out 3-pixels-by-52-pixels image data from the source memory 10. More specifically, the image data are read out up to (N-1, 176), (N, 176), and (N+1, 176) in the order of (N-1, 125), (N, 125), (N+1, 125), (N-1, 126), (N, 126), (N+1, 126), (N-1, 127), (N, 127), (N+1, 127), (N-1, 128), (N, 128), (N+1, 128), and so on.

[0224] The data readout unit 120 feeds the read image data (N-1, 125) etc. in sequence into the filtering circuit 3.

[0225] As illustrated in Fig. 7 (a), at cycle "CY1", the shift register 30 feeds nine pieces of serially aligned image data (N-1, 125) to (N+1, 127) into the filter calculator 31 in parallel.

[0226] The filter calculator 31 filters target image data subject to filtering (N, 126) using nine pieces of image data (N-1, 125) to (N+1, 127), thereby providing filtered image data (FN, 126) as shown in Fig. 7 (b).

[0227] As illustrated in Fig. 7 (c), the data write unit 121 writes the filtered image data (FN, 126) to the destination memory 11 in accordance with effective data-indicating signal "SE" from the filter calculator 31.

[0228] As shown in Fig. 7 (a), at the next cycle "CY2", the shift register 30 feeds nine pieces of image data (N, 125) to (N-1, 128) in parallel.

[0229] At the following cycle "CY3", the shift register 30 feeds nine pieces of image data (N+1, 125) to (N, 128) in parallel.

[0230] At cycle "CY4", the shift register 30 feeds nine pieces of image data (N-1, 126) to (N+1, 128) in parallel.

[0231] The filter calculator 31 filters target image data subject to filtering (N, 127) using the nine pieces of image data (N-1, 126) to (N+1, 128), thereby providing filtered image data (FN, 127) as shown in Fig. 7 (b).

[0232] As illustrated Fig. 7 (c), the data write unit 121 writes filtered image data (FN, 127) to the destination memory 11 in accordance with effective data-indicating signal "SE" from the filter calculator 31.

[0233] The above steps are repeated until filtering end signal "Se" enters the filtering circuit 3 from the data readout unit 120.

[0234] More specifically, as shown in Fig. 7 (a), the filtering circuit 3 repeats the filtering fifty times (where each time consists of three cycles), thereby providing fifty pieces of filtered image data (FN, 126) to (FN, 175). As a result, the destination memory 11 stores the filtered image data of (FN, 126) to (FN, 175).

[0235] As illustrated in Fig. 7 (b), the filter calculator 31 feeds and non-filters two pieces of image data (N, 125) and (N, 176) among the image data in the source memory 10. Image data (N, 125) and (N, 176) are located in the central line of the image data in the source memory 10 at both ends of the central line.

[0236] The data write unit 121 writes the non-filtered image data (N, 125), (N, 176) to the destination memory 11 in accordance with effective data-indication signal "SE" as shown in Fig. 7 (c).

[0237] Consequently, as illustrated Fig. 3 (c), the destination memory 11 stores the filtered image data of (FN, 126) to (FN, 175) and the non-filtered image data of (N,

125), (N, 176).

[0238] The image data of (FN, 126) to (FN, 175), (N, 125), and (N, 176) are transmitted to the image memory 2.

[0239] As a result, the image memory 2 stores: filtered image data (FN, 2) to (FN, 63) and non-filtered image data (N, 1), (N, 64); filtered image data (FN, 64) to (FN, 125) and non-filtered image data (N, 63), (N, 126); and filtered image data (FN, 126) to (FN, 175) and non-filtered image data (N, 125), and (N, 176).

[0240] The processor 1 removes unnecessary image data (N, 64), (N, 63), (N, 126), and (N, 125), thereby forming line data that consists of image data from (N, 1), (FN, 2) to (FN, 175), and (N, 176).

[0241] In this way, an image output circuit (not shown) feeds a subsequently created series of line data from the image memory 2 to a display device (not shown), thereby displaying the line data on the display device (not shown).

[0242] As described above, the image processor according to the present embodiment divides three-lines image data into partitions in accordance with the storage capacity of the source memory 10 to place the partitions into the source memory 10 in order to perform filtering calculation.

[0243] This feature eliminates a buffer for three lines (see a line buffer 502 of Fig. 25), to which image data are transferred from the image memory 2 to filter the image data, and a buffer for a single line (see a line buffer 504 of Fig. 25) to store filtered image data. This means that a reduced area of an image processor is achievable.

[0244] A prior art image processor reads out image data for each gang of three pixels-by-three pixels to filter the image data (see Figs. 27 and 28). Different from the prior art image processor, the image processor according to the present embodiment is operable to sequentially read out several pieces of image data from the source memory 10. This feature makes it feasible to filter several pieces of target image data subject to filtering without the need to read out the same image data several times. As a result,

the target image data subject to filtering can be filtered in a shorter time.

[0245] The image processor according to the present embodiment is operable to filter the target image data subject to filtering without providing a fixed number of pixels in a line. This means that the target image data subject to filtering can be filtered independently of an image size.

[0246] The present embodiment illustrates the QCIF-sized image as a target to be filtered. However, the present embodiment is not limited thereto.

[0247] Assume that "p"-lines-by-"q"-pixels image data are to be filtered, and further that each gang of "P"-pixels-by-"Q"-pixels is filtered. The "p"-lines-by-"q"-pixels image data is divided into several partitions in accordance with the storage capacity of the source memory 10. "P"-pixels-by-"Y"-pixels image data is transmitted to the source memory 10. As a result, filtering calculation is achievable.

[0248] (Second embodiment)

[0249] An image processor according to a second embodiment includes a filtering circuit 4 substituted for the filtering circuit 3 of Fig. 1. The filtering circuit 4 is discussed below. The image processor according to the present embodiment is similar in construction to the image processor of Fig. 1 except for the filtering circuit just discussed above.

[0250] Figs. 8(a), 8(b), and 8(c) are illustrations showing a flow of filtering steps. Fig. 8(a) is an illustration showing a flow of the first filtering. Fig. 8(b) is an illustration showing a flow of the second filtering. Fig. 8(c) is an illustration showing a flow of the third filtering. Fig. 8 illustrates the same reference characters on components similar to those shown in Fig. 1.

[0251] Fig. 9 is a block diagram illustrating the filtering circuit 4. As shown in Fig. 9, the filtering circuit 4 includes a data output control unit 80 other than the same components as those of the filtering circuit 3 of Fig. 4.

[0252] The data output control unit 80 includes a register 33 and an edgewise pixel

processor 32. The edgewise pixel processor 32 processes edgewise pixels.

[0253] The edgewise pixel processor 32 sometimes feeds image data designated as "\$". The edgewise pixel processor 32 feeds filtered image data as well as non-filtered image data as discussed later.

[0254] Figs. 10(a), 10(b), 10(c), 10(d), and 10(e) are time charts illustrating the first filtering. Fig. 10(a) is an illustration showing, for each cycle, image data provided to a filter calculator 31 of Fig. 9. Fig. 10(b) is an illustration showing image data fed from the filter calculator 31. Fig. 10(c) is an illustration showing an effective data-indicating signal from the filter calculator 31. Fig. 10 (d) is an illustration showing image data written by a data write unit 121 to a destination memory 11. Fig. 10 (e) is an illustration showing a write-indicating signal from the edgewise pixel processor 32.

[0255] Figs. 11(a), 11(b), 11(c), 11(d), and 11(e) are time charts illustrating the second filtering. Fig. 11(a) is an illustration showing, for each cycle, image data sent to the filter calculator 31. Fig. 11(b) is an illustration showing image data fed from the filter calculator 31. Fig. 11(c) is an illustration showing an effective data-indicating signal from the filter calculator 31. Fig. 11 (d) is an illustration showing image data written by the data write unit 121 to the destination memory 11. Fig. 11(e) is an illustration showing a write-indicating signal from the edgewise pixel processor 32.

[0256] Figs. 12(a), 12(b), 12(c), 12(d), and 12 (e) are time charts illustrating the third filtering. Fig. 12(a) is an illustration showing, for each cycle, image data sent to the filter calculator 31. Fig. 12(b) is an illustration showing image data fed from the filter calculator 31. Fig. 12(c) is an illustration showing an effective data-indicating signal from the filter calculator 31. Fig. 12(d) is an illustration showing image data written by the data write unit 121 to the destination memory 11. Fig. 12(e) is an illustration showing a write-indicating signal from the edgewise pixel processor 32.

[0257] The filtering steps are now described in the order of the first, second, and third filtering, in which the source memory 10 of Fig. 1 is adapted to store

3-lines-by-64-pixels image data at maximum.

[0258] Assume that each gang of three-pixels-by-three-pixels image data is filtered, and further that a pixel located at the center of three pixels-by-three pixels has target image data subject to filtering.

[0259] The first filtering is discussed with reference to Fig. 1, Fig. 8(a), Fig. 9, and Figs. 10(a) to 10(e).

[0260] Steps taken until the filter calculator 31 of Fig. 9 feeds filtered image data D# and effective data-indicating signal "SE" are the same as the steps (see Figs. 5(a) to 5(c)) taken until the filter calculator 31 of Fig. 4 according to the previous embodiment feeds filtered image data D# and effective data-indicating signal "SE". Therefore, descriptions related to the above steps are omitted.

[0261] However, before starting the first filtering, mode signal "SM" sets mode information "M" into the register 33 of Fig. 9. Mode information "M" prohibits feeding a rightmost piece of image data. Details of this step are discussed later.

[0262] As illustrated in Fig. 10(b), the edgewise pixel processor 32 receives image data (N, 1), (FN, 2) to (FN, 63), and (N, 64) from the filter calculator 31 in accordance with effective data-indicating signal "SE".

[0263] As seen from Fig. 10(d), the edgewise pixel processor 32 feeds required image data (N, 1), (FN, 2) to (FN, 63) among the entered image data (N, 1), (FN, 2) to (FN, 63), and (N, 64) into the data write unit 121 in accordance with mode information "M" from the register 33.

[0264] As seen from Fig. 10(d), the edgewise pixel processor 32 prohibits feeding unnecessary image data (N, 64) in accordance with mode information "M".

[0265] Image data (FN, 2) to (FN, 63) among required image data (N, 1), (FN, 2) to (FN, 63) are filtered image data.

[0266] As shown in Fig. 10(e), the edgewise pixel processor 32 feeds write-indicating signal "SW", as well as the required image data (N, 1), (FN, 2) to (FN, 63), into the

data write unit 121.

[0267] The data write unit 121 writes required image data (N, 1), (FN, 2) to (FN, 63) in sequence to the destination memory 11 in accordance with write-indicating signal "SW" from the edgewise pixel processor 32.

[0268] The data write unit 121 writes the above image data when the write-indicating signal "SW" is at the level of "H (high)".

[0269] Referring to Fig. 8 (a), as a result of the above edgewise pixel processing, only the required image data (N, 1), (FN, 2) to (FN, 63), not the unnecessary image data (N, 64), are written to the destination memory 11.

[0270] Image data (N, 1), (FN, 2) to (FN, 63) written to the destination memory 11 are transmitted to the image memory 2.

[0271] The second filtering is now discussed with reference to Fig. 1, Fig. 8(b), Fig. 9, and Figs. 11(a) to 11(e).

[0272] Steps taken until the filter calculator 31 of Fig. 9 feeds filtered image data D# and effective data-indicating signal "SE" are the same as the steps (see Figs. 6(a) to 6(c)) taken until the filter calculator 31 of Fig. 4 according to the previous embodiment feeds filtered image data D# and effective data-indicating signal "SE". Therefore, descriptions related to the above steps are omitted.

[0273] However, before starting the second filtering, mode signal "SM" sets mode information "M" into the register 33 of Fig. 9. Mode information "M" prohibits feeding rightmost and leftmost pieces of image data. Details of this step are discussed later.

[0274] As illustrated in Fig. 11(b), the edgewise pixel processor 32 receives image data (N, 63), (FN, 64) to (FN, 125), and (N, 126) from the filter calculator 31 in accordance with effective data-indicating signal "SE".

[0275] As seen from Fig. 11(d), the edgewise pixel processor 32 feeds required image data (FN, 64) to (FN, 125) among the entered image data (N, 63), (FN, 64) to (FN, 125), and (N, 126) into the data write unit 121 in accordance with mode information

"M" from the register 33. The second filtering differs in mode information "M" from the first filtering.

[0276] As seen from Fig. 11(d), the edgewise pixel processor 32 prohibits feeding unnecessary image data (N, 63) and (N, 126) in accordance with mode information "M".

[0277] The required image data (FN, 64) to (FN, 125) are filtered image data.

[0278] As shown in Fig. 11(e), the edgewise pixel processor 32 feeds write-indicating signal "SW", as well as the required image data (FN, 64) to (FN, 125), into the data write unit 121.

[0279] The data write unit 121 writes the required image data (FN, 64) to (FN, 125) in sequence to the destination memory 11 in accordance with write-indicating signal "SW" from the edgewise pixel processor 32.

[0280] Referring to Fig. 8(b), as a result of the above edgewise pixel processing, the only the required image data (FN, 64) to (FN, 125), not the unnecessary image data (N, 63) and (N, 126), are written to the destination memory 11.

[0281] The image data (FN, 64) to (FN, 125) written to the destination memory 11 are transmitted to the image memory 2.

[0282] The third filtering is now discussed with reference to Fig. 1, Fig. 8(c), Fig. 9, and Figs. 12(a) to 12(e).

[0283] Steps taken until the filter calculator 31 of Fig. 9 feeds filtered image data D# and effective data-indicating signal "SE" are the same as the steps (see Figs. 7(a) to 7(c)) taken until the filter calculator 31 of Fig. 4 according to the previous embodiment feeds filtered image data D# and effective data-indicating signal "SE". Therefore, descriptions related to the above steps are omitted.

[0284] However, mode signal "SM" sets mode information "M" into the register 33 of Fig. 9 before the third filtering starts. Mode information "M" prohibits feeding a leftmost piece of image data. Details of this step are discussed later.

[0285] As illustrated in Fig. 12(b), the edgewise pixel processor 32 receives image data (N, 125), (FN, 126) to (FN, 175), and (N, 176) from the filter calculator 31 in accordance with effective data-indicating signal "SE".

[0286] As seen from Fig. 12(d), the edgewise pixel processor 32 feeds required image data (FN, 126) to (FN, 175), and (N, 176) among the entered image data (N, 125), (FN, 126) to (FN, 175), and (N, 176) in accordance with mode information "M" from the register 33. The third filtering differs in mode information "M" from the second filtering.

[0287] As seen from Fig. 12(d), the edgewise pixel processor 32 does not feed unnecessary image data (N, 125) in accordance with mode information "M".

[0288] The image data (FN, 126) to (FN, 175) among the required image data (FN, 126) to (FN, 175), and (N, 176) are filtered image data.

[0289] As shown in Fig. 12(e), the edgewise pixel processor 32 feeds write-indicating signal "SW", as well as the required image data (FN, 126) to (FN, 175), and (N, 176), into the data write unit 121.

[0290] The data write unit 121 writes the required image data (FN, 126) to (FN, 175), (N, 176) in sequence to the destination memory 11 in accordance with write-indicating signal "SW" from the edgewise pixel processor 32.

[0291] Referring to Fig. 8(c), as a result of the above edgewise pixel processing, only the required image data (FN, 126) to (FN, 175), (N, 176), not the unnecessary image data (N, 125), are written to the destination memory 11.

[0292] The image data (FN, 126) to (FN, 175), and (N, 176) written to the destination memory 11 are transmitted to the image memory 2.

[0293] As a result, the image memory 2 of Fig. 1 stores line data that consist of image data (N, 1), (FN, 2) to (FN, 175), and (N, 176).

[0294] In this way, an image output circuit (not shown) feeds a successively created series of line data into a display device (not shown) from the image memory 2, thereby

displaying the line data on the display device (not shown).

[0295] The following discusses mode information "M" as illustrated in Fig. 9.

[0296] As shown in Fig. 9, the register 33 retains mode information "M" according to the entered mode signal "SM".

[0297] Mode information "M" is free to change according to mode signal "SM".

[0298] In the present embodiment, mode information "M" is set before the first filtering to process edgewise pixels during the first filtering.

[0299] Mode information "M" is again set after the first filtering to process the edgewise pixels during the second filtering.

[0300] Mode information "M" is again set after the second filtering to process edgewise pixels during the third filtering.

[0301] Mode signal "SM" is sent out by, e.g., a command from the processor 1.

[0302] Details on mode information "M" of Fig. 9 are now discussed.

[0303] Figs. 13(a) to 13(d) are descriptive illustrations showing mode information "M" settable in the register 33 of Fig. 9. Fig. 13(a) is a descriptive illustration showing mode information "M" of mode 2'b00. Fig. 13(b) is a descriptive illustration showing mode information "M" of mode 2'b01. Fig. 13(c) is a descriptive illustration showing mode information "M" of mode 2'b10. Fig. 13(d) is a descriptive illustration showing mode information "M" of mode 2'b11.

[0304] Figs. 13(a), 13(b), 13(c), and 13(d) illustrate the filtering using three-pixels-by-six-pixels image data 1 to 18 that are stored in the source memory 10 of Fig. 1.

[0305] The filter calculator 31 of Fig. 9 feeds image data 2, 5, 8, 11, 14, and 17 into the edgewise pixel processor 32.

[0306] Mode 2'b00 is now discussed with reference to Fig. 13(a). Mode 2'b00 shows that rightmost and leftmost pieces of image data 2, 17 among image data 2, 5, 8, 11, 14, and 17 from the filter calculator 31 are non-filtered image data required for display;

and the remaining image data 5, 8, 11, and 14 are filtered image data.

[0307] When mode signal "SM" sets mode information "M" of mode 2'b00 into the register 33, then the edgewise pixel processor 32 feeds all of the entered image data 2, 5, 8, 11, 14, and 17 as well as write-indicating signal "SW". The write-indicating signal "SW" writes all of the entered image data 2, 5, 8, 11, 14, and 17 to the destination memory 11.

[0308] Fig. 13(a) to Fig. 13(c) name the non-filtered image data required for display as "through-data".

[0309] Mode 2'b00 is set when 3-lines-by-176-pixels image data is transmitted to the source memory 10 to filter target image data subject to filtering, provided that a QCIF image is to be processed, and further that each gang of three pixels-by-three pixels is to be filtered.

[0310] Mode 2'b01 is now discussed with reference to Fig. 13(b). Mode 2'b01 shows that image data 2 among image data 2, 5, 8, 11, 14, and 17 from the filter calculator 31 is non-filtered image data required for display; image data 5, 8, 11, and 14 are filtered image data; and the remaining image data 17 is unnecessary image data.

[0311] When mode signal "SM" sets mode information "M" of mode 2'b01 into the register 33, then the edgewise pixel processor 32 feeds image data 2, 5, 8, 11, and 14 among the entered image data 2, 5, 8, 11, 14, and 17, but does not feed image data 17.

[0312] The edgewise pixel processor 32 feeds write-indicating signal "SW" to the data write unit 121. The write-indicating signal "SW" allows the data write unit 121 to write only image data 2, 5, 8, 11, and 14 to the destination memory 11.

[0313] Mode 2'b01 is set when, e.g., a rightmost piece of image data (N, 64) is not fed as illustrated in Fig. 10(d).

[0314] As a result, the register 33 is set into mode 2'b01 during the first filtering.

[0315] Mode 2'b10 is now discussed with reference to Fig. 13(c). Mode 2'b10 shows that image data 2 among image data 2, 5, 8, 11, 14, and 17 from the filter calculator 31

is unnecessary image data; image data 5, 8, 11, and 14 are filtered image data; and the remaining image data 17 is non-filtered image data required for display.

[0316] When mode signal "SM" sets mode information "M" of mode 2'b10 into the register 33, then the edgewise pixel processor 32 feeds image data 5, 8, 11, 14, and 17 among image data 2, 5, 8, 11, 14, and 17, but does not feed image data 2.

[0317] The edgewise pixel processor 32 feeds write-indicating signal "SW" to the data write unit 121. The write-indicating signal "SW" allows the data write unit 121 to write only image data 5, 8, 11, 14, and 17 to the destination memory 11.

[0318] Mode 2'b10 is set when, e.g., a leftmost piece of image data (N, 125) is not fed as illustrated in Fig. 12(d).

[0319] As a result, the register 33 is set into mode 2'b10 during the third filtering.

[0320] Mode 2'b11 is now discussed with reference to Fig. 13(d). Mode 2'b11 shows that image data 2, 17 among image data 2, 5, 8, 11, 14, and 17 from the filter calculator 31 are unnecessary image data; and image data 5, 8, 11, and 14 are filtered image data,

[0321] When mode signal "SM" sets mode information "M" of mode 2'b11 into the register 33, then the edgewise pixel processor 32 feeds image data 5, 8, 11, and 14 among image data 2, 5, 8, 11, 14, and 17, but does not feed image data 2 and 17.

[0322] The edgewise pixel processor 32 feeds write-indicating signal "SW" to the data write unit 121. The write-indicating signal "SW" allows the data write unit 121 to write only image data 5, 8, 11, and 14 to the destination memory 11.

[0323] Mode 2'b11 is set when, e.g., rightmost and leftmost pieces of image data (N, 63), (N, 126) are not fed as illustrated in Fig. 11(d).

[0324] As a result, the register 33 is set into mode 2'b11 during the second filtering.

[0325] The following discusses mode information "M" with reference to a specific example.

[0326] Figs. 14(a), 14(b), and 14(c) illustrate how mode information "M" is used. Fig. 14(a) illustrates the use of mode 2'b01. Fig. 14(b) illustrates the use of mode 2'b11.

Fig. 14(c) illustrates the use of mode 2'b10.

[0327] Figs. 14(a), 14(b), and 14(c) show, as an illustration, how three-pixels-by-six-pixels image data are filtered.

[0328] As illustrated in Figs. 14 (a) to 14 (c), assume that the image memory 2 of Fig. 1 stores image data 1 to 18.

[0329] As shown in Fig. 14 (a), assume that image data 1 to 9 are transmitted to the source memory 10 of Fig. 1.

[0330] At this time, the register 33 is set into mode 2'b01. As a result, the edgewise pixel processor 32 feeds only image data 2 and 5 among the entered image data 2, 5, and 8 from the filter calculator 31.

[0331] As illustrated in Fig. 14 (b), assume that image data 4 to 15 are transmitted to the source memory 10.

[0332] At this time, the register 33 is set into mode 2'b11. As a result, the edgewise pixel processor 32 feeds only image data 8 and 11 among the entered image data 5, 8, 11, and 14 from the filter calculator 31.

[0333] As shown in Fig. 14 (c), assume that image data 10 to 18 are transmitted to the source memory 10.

[0334] At this time, the register 33 of Fig. 9 is set into mode 2'b10. As a result, the edgewise pixel processor 32 feeds only image data 14 and 17 among the entered image data 11, 14, and 17 from the filter calculator 31.

[0335] As described above, the image processor according to the present embodiment is possible to prepare line data in a reduced time because a gang of filtered image data is free of unnecessary image data (N, 64), (N, 63), (N, 126), and (N, 125), as illustrated in Fig. 8. The unnecessary image data (N, 64), (N, 63), (N, 126), and (N, 125) are produced in the course of filtering the target image data subject to filtering.

[0336] The image processor according to the present embodiment includes the construction of the image processor according to the previous embodiment, and

consequently provides beneficial effects similar to those achieved in the previous embodiment.

[0337] (Third embodiment)

[0338] An image processor according to a third embodiment includes a filtering circuit 5 substituted for the filtering circuit 3 of Fig. 1. The filtering circuit 5 is discussed below. The image processor according to the present embodiment is similar to those of the image processor of Fig. 1 except for the filtering circuit as just discussed above.

[0339] Fig. 15 is a block diagram illustrating the filtering circuit 5. As shown in Fig. 15, the filtering circuit 5 includes a pixel-retaining buffer unit 50, selectors 52, 53, and a selector controller 51, other than components of the filtering circuit 4 of Fig. 9.

[0340] Filtering is now discussed in brief.

[0341] Fig. 16 is a schematic illustration showing how the image processor according to the present embodiment is operable to filter target image data subject to filtering. Fig. 16 is similar in notation to Fig. 2.

[0342] As illustrated in Fig. 16, three-pixels-by-five-pixels image data 101 obtained by dividing 3-lines-by-176-pixels image data 100 is transmitted to the source memory 10 from the image memory 2 of Fig. 1.

[0343] The source memory 10 feeds the image data 101 in sequence to the filtering circuit 5 for each piece of image data.

[0344] The filtering circuit 5 filters a piece of target image data subject to filtering using the given three-pixels-by-three-pixels image data.

[0345] In this way, the filtering circuit 5 filters all three pieces of target image data subject to filtering that are stored in the image data 101.

[0346] When completing the present filtering, then the filtering circuit 5 places image data 109 into the pixel-retaining buffer unit 50 for use at the time of the next filtering.

[0347] Subsequently, three-pixels-by-four-pixels image data 106 obtained by dividing 3-lines-by-176-pixels image data 100 is transmitted to the source memory 10 from the

image memory 2.

[0348] The filtering circuit 5 filters target image data subject to filtering using the image data 106 in the source memory 10 and the image data 109 in the pixel-retaining buffer 50.

[0349] The above steps are repeated to filter all pieces of target image data subject to filtering that are stored in 3-lines-by-176-pixels image data 100. As a result, a line of filtered image data 105 is achieved.

[0350] As previously discussed, when filtering the image data 101 is completed, then the filtering circuit 5 according to the present embodiment puts the image data 109 among the image data 101 into the pixel-retaining buffer unit 50 in order to use the image data 109 at the time of the following filtering.

[0351] This feature provides a smaller number of times of transferring image data from the image memory 2 to the source memory 10 and a smaller number of times of transferring image data from the source memory 10 to the filtering circuit 5.

[0352] The following discusses details of the filtering with reference to Figs. 1, 15, 17, and 18, in which the source memory 10 of Fig. 1 is adapted to store 3-lines-by-64-pixels image data at maximum.

[0353] Assume that each gang of three-pixels-by-three-pixels image data is filtered, and further that a pixel located at the center of three pixels-by-three pixels has target image data subject to filtering.

[0354] Figs. 17(a), 17(b), and 17(c) are illustrations showing a flow of filtering steps. Fig. 17(a) is an illustration showing a flow of first filtering. Fig. 17(b) is an illustration showing a flow of second filtering. Fig. 17(c) is an illustration showing a flow of third filtering. Fig. 17 illustrates the same reference numerals on components similar to those described in Fig. 1.

[0355] Figs. 18(a), 18(b), and 18(c) are time charts showing characteristics of the filtering according to the present embodiment. Fig. 18(a) is an illustration showing

how image data is transferred from a shift register 30 to the pixel-retaining buffer unit 50 at the end of the first filtering. Fig. 18(b) is an illustration showing when the second filtering is started. Fig. 18(c) is an illustration showing image data written to a destination memory 11 from a data write unit 121.

[0356] The first filtering is similar to that according to the second embodiment as illustrated in Figs. 10(a) to 10(e). Therefore, descriptions related thereto are omitted.

[0357] However, the first filtering is followed by additional steps as discussed below when the first filtering is completed.

[0358] As illustrated in Fig. 18(a), at the last cycle "CY184" of the first filtering, flip-flops FF1 to FF9 of the shift register 30 feeds image data (N-1, 62) to (N+1, 64), respectively.

[0359] Image data (N-1, 63), (N, 63), (N+1, 63), (N-1, 64), (N, 64), and (N+1, 64) among the above image data (N-1, 62) to (N+1, 64) are placed into the pixel-retaining buffer unit 50 for use at the time of the second filtering.

[0360] More specifically, the pixel-retaining buffer unit 50 stores image data (N-1, 63), (N, 63), (N+1, 63), (N-1, 64), (N, 64), and (N+1, 64) that are supplied at the last cycle "CY184" from flip-flops FF4, FF5, FF6, FF7, FF8, and FF9, respectively.

[0361] The second filtering is now described. After the first filtering ends, mode signal "SM" sets up mode information "M" (mode 2'b11 of Fig. 13 (d)) that prohibits feeding rightmost and leftmost pieces of image data.

[0362] As illustrated in Fig. 17(b), image data (N-1, 65), (N, 65), (N+1, 65) to (N-1, 126), (N, 126), and (N+1, 126) are transmitted to the source memory 10 from the image memory 2.

[0363] At this time, image data (N-1, 63), (N, 63), (N+1, 63), (N-1, 64), (N, 64), and (N+1, 64) are not transmitted to the source memory 10. Instead, the second filtering employs image data (N-1, 63), (N, 63), (N+1, 63), (N-1, 64), (N, 64), and (N+1, 64), all of which are retained in pixel-retaining buffer unit 50.

[0364] As shown in Fig. 18 (b), at the first cycle "CY#1" of the second filtering, the flip-flop FF 9 feeds image data (N-1, 65) from the source memory 10.

[0365] At cycle "CY#1", the flip-flops FF3, FF6 feed image data (N-1, 63), (N-1, 64) from the pixel-retaining buffer unit 50, respectively.

[0366] At the following cycle "CY#2", flip-flop FF 9 feeds image data (N, 65) from the source memory 10, while flip-flop FF 8 feeds image data (N-1, 65) that is shifted from flip-flop FF 9.

[0367] At cycle "CY#2", flip-flops FF3, FF6 feeds image data (N, 63), (N, 64) from the pixel-retaining buffer unit 50, respectively, while flip-flops FF2, FF5 feeds image data (N-1, 63), (N-1, 64) that are shifted from flip-flops FF3, FF6, respectively.

[0368] At the following cycle "CY#3", flip-flop FF 9 feeds image data (N+1, 65) from the source memory 10, while flip-flops FF7, FF8 feed image data (N-1, 65), (N, 65) that are shifted from flip-flops FF8, FF9, respectively.

[0369] At cycle "CY#3", flip-flops FF3, FF6 feeds image data (N+1, 63), (N+1, 64) from the pixel-retaining buffer unit 50, respectively, while flip-flops FF1, FF2, FF4, and FF5 feed image data (N-1, 63), (N, 63), (N-1,64), and (N, 64) that are shifted from flip-flops FF2, FF3, FF5, and FF6, respectively.

[0370] At this time, as shown in Fig. 18 (c), the first filtered image data (FN, 64) is fed. Thereafter, usual shift actions similar to those according to the second embodiment are provided to filter target image data subject to filtering.

[0371] Similar to the first filtering, image data for use at the time of the third filtering are held in the pixel-retaining buffer unit 50 at the end of the second filtering.

[0372] More specifically, when the second filtering is terminated, then the pixel-retaining buffer unit 50 retains image data (N-1, 125), (N, 125), (N+1,125), (N-1, 126), (N, 126), and (N+1,126) from flip-flops FF4, FF5, FF6, FF7, FF8, and FF9, respectively.

[0373] The third filtering is now described. After the end of the second filtering, mode

signal "SM" sets up mode information "M" (mode 2'b10 of Fig. 13(c)) that prohibits feeding a leftmost piece of image data.

[0374] As shown in Fig. 17(c), image data (N-1, 127), (N, 127), (N+1, 127) to (N-1, 176), (N, 176), and (N+1, 176) are transmitted to the source memory 10 from the image memory 2.

[0375] At this time, image data (N-1, 125), (N, 125), (N+1, 125), (N-1, 126), (N, 126), and (N+1, 126) are not transmitted to the source memory 10. Instead, the third filtering employs image data (N-1, 125), (N, 125), (N+1, 125), (N-1, 126), (N, 126), and (N+1, 126) that are held in the pixel-retaining buffer unit 50.

[0376] In the third filtering, first to third cycles are similar to those in the second filtering. More specifically, the shift register 30 feeds image data from the source memory 10 and the pixel-retaining buffer unit 50 to filter target image data subject to filtering. The filtering at the fourth or greater cycles is similar to that according to the second embodiment.

[0377] The following discusses, with reference to Fig. 15, details of image data transfer to the pixel-retaining buffer unit 50 and details of image data transfer from the pixel-retaining buffer unit 50 to the shift register 30.

[0378] Mode information "M" currently held in the register 33 is sent to the selector controller 51.

[0379] The selector controller 51 produces selection signal "SC" based on mode information "M". The selection signal SC is provided to selectors 52 and 53.

[0380] The selector 52 receives image data from the pixel-retaining buffer unit 50 and image data from flip-flop FF 4.

[0381] The selector 52 selects either the image data from the pixel-retaining buffer unit 50 or that from flip-flop FF 4 in accordance with selection signal "SC". The selector 52 feeds the selected image data into flip-flop FF 3.

[0382] The selector 53 receives image data from the pixel-retaining buffer unit 50 and

image data from flip-flop FF 7.

[0383] The selector 53 selects either the image data from the pixel-retaining buffer unit 50 or that from flip-flop FF 7 in accordance with selection signal "SC". The selector 53 feeds the selected image data into the flip-flop FF 6.

[0384] The above steps are now described with reference to Figs. 18(a) and 18(b). As illustrated in Fig. 18(a), at cycle "CY184" of the first filtering, image data (N-1, 63), (N, 63), (N+1, 63), (N-1, 64), (N, 64), and (N+1, 64) from flip-flops FF4, FF5, FF6, FF7, FF8, and FF9 are held in the pixel-retaining buffer unit 50.

[0385] As shown in Fig. 18(b), at cycle "CY#1" of the second filtering, flip-flops FF3, FF6 feed image data (N-1, 63), (N-1, 64), respectively.

[0386] The selector 52 selects image data (N-1, 63) from the pixel-retaining buffer unit 50 before cycle "CY#1" in accordance with selection signal "SC", and then feeds the selected image data (N-1, 63) into flip-flop FF 3.

[0387] The selector 53 selects image data (N-1, 64) from the pixel-retaining buffer unit 50 before cycle "CY#1" in accordance with selection signal "SC", and then feeds the selected image data (N-1, 64) into flip-flop FF 6.

[0388] As shown in Fig. 18 (b), at cycle "CY#2", flip-flops FF3, FF6 feed image data (N, 63), (N, 64), respectively.

[0389] The selector 52 selects image data (N, 63) from the pixel-retaining buffer unit 50 before cycle "CY#2" in accordance with selection signal "SC", and then feeds the selected image data (N, 63) into flip-flop FF 3.

[0390] The selector 53 selects image data (N, 64) from the pixel-retaining buffer unit 50 before cycle "CY#2" in accordance with selection signal "SC", and then feeds the selected image data (N, 64) into flip-flop FF 6.

[0391] As illustrated in Fig. 18 (b), at cycle "CY#3", flip-flops FF3, FF6 feeds image data (N+1, 63), (N+1, 64), respectively.

[0392] The selector 52 selects image data (N+1, 63) from the pixel-retaining buffer

unit 50 before cycle "CY#3" in accordance with selection signal "SC", and then feeds the selected image data (N+1, 63) into flip-flop FF 3.

[0393] The selector 53 selects image data (N+1, 64) from the pixel-retaining buffer unit 50 before cycle "CY#3" in accordance with selection signal "SC", and then feeds the selected image data (N+1, 64) into flip-flop FF 6.

[0394] Thereafter, the selectors 52, 53 select, in accordance with selection signal "SC", respective pieces of image data that flip-flops FF4, FF7 feed.[0394]

[0395] As previously discussed, the image processor according to the present embodiment is operable to retain image data required for use at the time of the next filtering when the present filtering is terminated. As a result, the retained image data are usable at the time of the next filtering. This feature provides a reduced number of times of transferring the data.

[0396] The image processor according to the present embodiment includes the construction of the image processor according to the second embodiment. Therefore, the image processor according to the present embodiment provides beneficial effects similar to those obtained in the second embodiment.

[0397] (Fourth embodiment)

[0398] An image processor according to a fourth embodiment includes a filtering circuit 6 substituted for the filtering circuit 3 of Fig. 1. The filtering circuit 6 is discussed below. The image processor according to the present embodiment is similar in construction to the image processor of Fig. 1 except for filtering circuit as just discussed above.

[0399] Fig. 19 is a block diagram illustrating the filtering circuit 6. Fig. 19 illustrates the same reference characters on components similar to those described in Fig. 9.

[0400] As illustrated in Fig. 19, the filtering circuit 6 includes a data output control unit 81 that is substituted for the data output control unit 80 of Fig. 9.

[0401] The data output control unit 81 includes a down counter 62, a coincidence

detector 61, and an edgewise pixel processor 32.

[0402] Pixel number-setting signal "SN" enters the down counter 62 and coincidence detector 61 in response to, e.g., instructions from the processor 1 before the filtering starts. As a result, the number of pixels for one line is set as initial value in the down counter 62 and coincidence detector 61.

[0403] The down counter 62 counts down each time when the edgewise pixel processor 32 feeds image data D\$.

[0404] More specifically, the down counter 62 provides a countdown each time when the edgewise pixel processor 32 feeds write-indicating signal "SW" having a "H"-level into the down counter 62.

[0405] The down counter 62 feeds countdown signal "SD" into the coincidence detector 61. Countdown signal "SD" shows the number of count.

[0406] The coincidence detector 61 compares the number of pixels, set up by pixel number-setting signal "SN", with the number of count that is expressed by countdown signal "SD". When the number of pixels is coincident with the number of count, then the coincidence detector 61 feeds coincidence signal "SA" having a "H"-level into the edgewise pixel processor 32.

[0407] The coincidence detector 61 feeds coincidence signal "SA" having "H"-level into the edgewise pixel processor 32 when countdown signal "SD" names the number of count as "1".

[0408] As a result, coincidence signal "SA" having "H"-level is sent out only before the first and last image data "D\$" are fed from the edgewise pixel processor 32.

[0409] The edgewise pixel processor 32 feeds the entered image data when coincidence signal "SA" having "H"-level enters the edgewise pixel processor 32, when coincidence signal "SA" having "L"-level, not filtering start signal "Ss", enters the edgewise pixel processor 32, and when coincidence signal "SA" having "L"-level, not filtering end signal "Se", enters the edgewise pixel processor 32.

[0410] The edgewise pixel processor 32 does not feed the entered image data when coincidence signal "SA" having "L"-level as well as filtering start signal "Ss" enters the edgewise pixel processor 32, and when coincidence signal "SA" having "L"-level as well as filtering end signal "Se" enters the edgewise pixel processor 32. Specific examples of the above are discussed later.

[0411] The edgewise pixel processor 32 feeds a write-indicating signal having "H"-level when coincidence signal "SA" having "H"-level enters the edgewise pixel processor 32, when coincidence signal "SA" having "L"-level, not filtering start signal "Ss", enters the edgewise pixel processor 32, and when coincidence signal "SA" having "L"-level, not filtering end signal "Se", enters the edgewise pixel processor 32.

[0412] The edgewise pixel processor 32 does not feed the write-indicating signal having "H"-level when coincidence signal "SA" having "L"-level as well as filtering start signal "Ss" enters the edgewise pixel processor 32, and when coincidence signal "SA" having "L"-level as well as filtering end signal "Se" enters the edgewise pixel processor 32. Specific examples of the above are discussed later.

[0413] Features of the filtering according to the present embodiment are now described in brief.

[0414] Fig. 20 is a schematic illustration showing how the image processor according to the present embodiment is operable to filter target image data subject to filtering.

[0415] Fig. 20 is similar in notation to Fig. 2.

[0416] As illustrated in Fig. 20, pursuant to the present embodiment, 176-pixels of one line are set in the down counter 62.

[0417] The down counter 62 counts downward by "1" every time when the edgewise pixel processor 32 feeds image data "D\$".

[0418] The data output control unit 81 feeds only required image data "\$" among the entered image data in accordance with information from the down counter 62.

[0419] In this way, the edgewise pixel-processing of 3-lines-by-176-pixels image data

is achievable only by at first setting the number of pixels for one line into the down counter 62.

[0420] According to the second embodiment, mode information M is set up three times in order to perform the edgewise pixel-processing of 3-lines-by-176-pixels image data.

[0421] Details of the filtering are now discussed.

[0422] The following description presupposes that the source memory 10 of Fig. 1 is possible to store 3-lines-by-64-pixels image data at maximum.

[0423] Each gang of three-pixels-by-three-pixels image data is filtered. A pixel located at the center of three pixels-by-three pixels has target image data subject to filtering.

[0424] Figs. 21(a), 21(b), and 21(c) are illustrations showing a flow of filtering steps. Fig. 21(a) is an illustration showing a flow of the first filtering. Fig. 21(b) is an illustration showing a flow of the second filtering. Fig. 21(c) is an illustration showing a flow of the third filtering.

[0425] Figs. 21(a), 21(b), and 21(c) provide the same reference characters on components similar to those described in Fig. 1.

[0426] Figs. 22(a), 22(b), 22(c), 22(d), 22(e), and 22(f) are time charts illustrating the first filtering. Fig. 22(a) is an illustration showing, for each cycle, image data sent to the filter calculator 31 of Fig. 19. Fig. 22(b) is an illustration showing the number of count provided by the down counter 62. Fig. 22(c) is an illustration showing image data fed from the filter calculator 31. Fig. 22(d) is an illustration showing an effective data-indicating signal from the filter calculator 31. Fig. 22(e) is an illustration showing image data written by a data write unit 121 to a destination memory 11. Fig. 22(f) is an illustration showing a write-indicating signal fed from the edgewise pixel processor 32.

[0427] Figs. 23(a), 23(b), 23(c), 23(d), 23(e), and 23(f) are time charts illustrating the second filtering. Fig. 23(a) is an illustration showing, for each cycle, image data sent to the filter calculator 31. Fig. 23(b) is an illustration showing the number of count provided by the down counter 62. Fig. 23(c) is an illustration showing image data fed

from the filter calculator 31. Fig. 23(d) is an illustration showing an effective data-indicating signal from the filter calculator 31. Fig. 23(e) is an illustration showing image data written by the data write unit 121 to the destination memory 11. Fig. 23(f) is an illustration showing a write-indicating signal from the edgewise pixel processor 32.

[0428] Figs. 24(a), 24(b), 24(c), 24(d), 24(e), and 24(f) are time charts illustrating the third filtering. Fig. 24(a) is an illustration showing, for each cycle, image data sent to the filter calculator 31. Fig. 24(b) is an illustration showing the number of count provided by the down counter 62. Fig. 24(c) is an illustration showing image data fed from the filter calculator 31. Fig. 24(d) is an illustration showing an effective data-indicating signal from the filter calculator 31. Fig. 24(e) is an illustration showing image data written by the data write unit 121 to the destination memory 11. Fig. 24(f) is an illustration showing a write-indicating signal from the edgewise pixel processor 32.

[0429] The first filtering is now described with reference to Fig. 1, Fig. 19, Fig. 21(a), and Figs. 22(a) to 22(f).

[0430] Steps taken until the filter calculator 31 feeds filtered image data "D#" and effective data-indicating signal "SE" are the same as those (see Fig. 5(a) to 5(c)) taken until the filter calculator 31 of Fig. 4 according to the first embodiment feeds filtered image data "D#" and effective data-indicating signal "SE".

[0431] As illustrated in Fig. 19, pixel number-setting signal "SN" sets "176"-pixels for one line into the down counter 62 and the coincidence detector 61 before the first filtering.

[0432] As shown in Fig. 22(b), the down counter 62 initially feeds countdown signal "SD" to the coincidence detector 61. Countdown signal "SD" expresses the number of count, "176".

[0433] At this time, the number of pixels, "176", set up by pixel number-setting signal

"SN" is coincident with the number of count, "176", expressed by countdown signal "SD". As a result, the coincidence detector 61 feeds coincidence signal "SA" having "H"-level into the edgewise pixel processor 32.

[0434] As shown in Fig. 22(e) and 22(f), the edgewise pixel processor 32 in receipt of coincidence signal "SA" feeds image data (N, 1) from the filter calculator 31 and write-indicating signal "SW" having "H"-level.

[0435] The edgewise pixel processor 32 feeds write-indicating signal "SW" having "H"-level into the down counter 62.

[0436] The down counter 62 counts down by one from initial value "176". As illustrated in Fig. 22(b), the down counter 62 feeds countdown signal "SD" representative of the number of count, "175", into the coincidence detector 61.

[0437] At this time, the number of pixels, "176", set up by pixel number-setting signal "SN" does not coincide with the number of count, "175", expressed by countdown signal "SD". As a result, the coincidence detector 61 feeds coincidence signal "SA" having "L"-level.

[0438] As shown in Fig. 22(e) and 22(f), the edgewise pixel processor 32 feeds filtered image data (FN, 2) from the filter calculator 31 and write-indicating signal "SW" having "H"-level.

[0439] As illustrated in Fig. 22(f), the edgewise pixel processor 32 feeds write-indicating signal "SW" having "H"-level into the down counter 62.

[0440] Thereafter, the coincidence detector 61 continues to feed coincidence signal "SA" having "L"-level. As illustrated in Fig. 22(e), the edgewise pixel processor 32 feeds filtered image data (FN, 3) to (FN, 63) from the filter calculator 31 into the data write unit 121.

[0441] As illustrated in Fig. 22(f), the edgewise pixel processor 32 feeds write-indicating signal "SW" having "H"-level into the data write unit 121 while feeding the filtered image data (FN, 3) to (FN, 63) into the data write unit 121.

[0442] A data readout unit 120 feeds filtering end signal "Se" into the edgewise pixel processor 32 after the edgewise pixel processor 32 feeds filtered image data (FN, 63) into the data write unit 121.

[0443] As illustrated in Fig. 22(e), the edgewise pixel processor 32 in receipt of filtering end signal "Se" and coincidence signal "SA" having "L"-level does not feed the last image data (N, 64) from the filter calculator 31.

[0444] As illustrated in Fig. 22(f), the edgewise pixel processor 32 in receipt of filtering end signal "Se" and coincidence signal "SA" having "L"-level does not feed write-indicating signal "SW" having "H"-level.

[0445] As a result, the down counter 62 does not count down, and the number of count at the end of the first filtering is "113".

[0446] Consequently, as illustrated in Fig. 21(a), only the required image data (N, 1), (FN, 2) to (FN, 63), not the unnecessary image data (N, 64), are written to the destination memory 11.

[0447] The image data (N, 1), (FN, 2) to (FN, 63) written to the destination memory 11 are transferred to the image memory 2.

[0448] The second filtering is now discussed with reference to Fig. 1, Fig. 19, Fig. 21(b), and Fig. 23(a) to 23(f).

[0449] Steps taken until the filter calculator 31 feeds filtered image data "D#" and effective data-indicating signal "SE" are the same as those (see Fig. 6(a) to 6(c)) taken until the filter calculator 31 of Fig. 4 according to the first embodiment feeds filtered image data "D#" and effective data-indicating signal "SE". Therefore, descriptions related thereto are omitted.

[0450] As illustrated in Figs. 23(c) and 23(d), the edgewise pixel processor 32 receives image data (N, 63), (FN, 64) to (FN, 125), and (N, 126) from the filter calculator 31 in accordance with effective data-indicating signal "SE".

[0451] At this time, the number of pixels, "176", set up by the pixel number-setting

signal "SN" as illustrated in Fig. 23(b) does not coincide with the number of count shown by countdown signal "SD". The coincidence detector 61 feeds coincidence signal "SA" having "L"-level into the edgewise pixel processor 32.

[0452] In order to start the second filtering, the data readout unit 120 feeds filtering start signal "Ss" into the edgewise pixel processor 32.

[0453] As shown in Fig. 23(e), the edgewise pixel processor 32 in receipt of filtering start signal "Ss" and coincidence signal "SA" having "L"-level does not feed image data (N, 63) that is initially fed from the filter calculator 31 into the edgewise pixel processor 32.

[0454] As shown in Fig. 23(f), the edgewise pixel processor 32 in receipt of filtering start signal "Ss" and coincidence signal "SA" having "L"-level does not feed write-indicating signal "SW" having "H"-level.

[0455] As a result, the down counter 62 performs no countdown. At this time, the number of count is still "113", as shown in Fig. 23(b).

[0456] Thereafter, coincidence signal "SA" having "L"-level, not filtering start signal "Ss", enters the edgewise pixel processor 32. As illustrated in Fig. 23(e), the edgewise pixel processor 32 feeds filtered image data (FN, 64) to (FN, 125) from the filter calculator 31.

[0457] As shown in Fig. 23(f), the edgewise pixel processor 32 feeds write-indicating signal "SW" having "H"-level to the data write unit 121 while feeding the filtered image data (FN, 64) to (FN, 125) into the data write unit 121.

[0458] The data readout unit 120 feeds filtering end signal "Se" into the edgewise pixel processor 32 after the edgewise pixel processor 32 feeds the filtered image data (FN, 125), as shown in Fig. 23 (e).

[0459] As illustrated in Fig. 23(e), the edgewise pixel processor 32 in receipt of filtering end signal "Se" and coincidence signal "SA" having "L"-level does not feed image data (N, 126) that is the last image data fed from the filter calculator 31.

[0460] As illustrated in Fig. 23(f), the edgewise pixel processor 32 in receipt of filtering end signal "Se" and coincidence signal "SA" having "L"-level does not feed write-indicating signal "SW" having "H"-level.

[0461] As a result, the down counter 62 performs no countdown, and the number of count at the end of the second filtering is "51".

[0462] Consequently, as illustrated in Fig. 21(b), only the required image data (FN, 64) to (FN, 125), not the unnecessary image data (N, 63) and (N, 126), are written to the destination memory 11,

[0463] The image data (FN, 64) to (FN, 125) written to the destination memory 11 are transmitted to the image memory 2.

[0464] The third filtering is now discussed with reference to Fig. 1, Fig. 19, Fig. 21(c), and Fig. 24(a) to 24(f).

[0465] Steps taken until the filter calculator 31 feeds filtered image data "D#" and effective data-indicating signal "SE" are the same as those (see Fig. 7(a) to 7(c)) taken until the filter calculator 31 of Fig. 4 according to the first embodiment feeds filtered image data "D#" and effective data-indicating signal "SE". Therefore, descriptions related thereto are omitted.

[0466] As illustrated in Figs. 24(c) and 24(d), the edgewise pixel processor 32 receives image data (N, 125), (FN, 126) to (FN, 175), and (N, 176) from the filter calculator 31 in accordance with effective data-indicating signal "SE".

[0467] At this time, the number of pixels, "176", set up by the pixel number-setting signal "SN" as illustrated in Fig. 24(b) does not coincide with the number of count shown by countdown signal "SD". The coincidence detector 61 feeds coincidence signal "SA" having "L"-level to the edgewise pixel processor 32.

[0468] In order to start the third filtering, the data readout unit 120 feeds filtering start signal "Ss" to the edgewise pixel processor 32.

[0469] As shown in Fig. 24(e), the edgewise pixel processor 32 in receipt of filtering

start signal "Ss" and coincidence signal "SA" having "L"-level does not feed image data (N, 125) that is the first image data fed into the edgewise pixel processor 32 from the filter calculator 31.

[0470] As shown in Fig. 24(f), the edgewise pixel processor 32 in receipt of filtering start signal "Ss" and coincidence signal "SA" having "L"-level does not feed write-indicating signal "SW" having "H"-level.

[0471] As a result, the down counter 62 performs no countdown. At this time, the number of count is still "51", as shown in Fig. 24(b).

[0472] Thereafter, coincidence signal "SA" having "L"-level, not filtering start signal "Ss", enters the edgewise pixel processor 32. As illustrated in Fig. 24(e), the edgewise pixel processor 32 feeds filtered image data (FN, 126) to (FN, 175) from the filter calculator 31.

[0473] As shown in Fig. 24(f), the edgewise pixel processor 32 feeds write-indicating signal "SW" having "H"-level to the data write unit 121 while feeding the filtered image data (FN, 126) to (FN, 175) into the data write unit 121.

[0474] The down counter 62 provides the number of count, "1", after the edgewise pixel processor 32 feeds filtered image data (FN, 175), as shown in Fig. 24(b).

[0475] As a result, countdown signal "SD" representative of the number of count, "1", enters the coincidence detector 61. The coincidence detector 61 feeds coincidence signal "SA" having "H"-level to the edgewise pixel processor 32.

[0476] As illustrated in Fig. 24(e), the edgewise pixel processor 32 in receipt of coincidence signal "SA" having "H"-level feeds image data (N, 126) that is the last image data coming from the filter calculator 31 to the edgewise pixel processor 32.

[0477] As illustrated in Fig. 24(f), the edgewise pixel processor 32 in receipt of coincidence signal "SA" having "H"-level feeds write-indicating signal "SW" having "H"-level.

[0478] Consequently, as illustrated in Fig. 21(c), only the required image data (FN,

126) to (FN, 175), and (N, 176), not the unnecessary image data (N, 125), are written to the destination memory 11.

[0479] The image data (FN, 126) to (FN, 175), and (N, 176) written to the destination memory 11 are transmitted to the image memory 2.

[0480] As described above, the image processor according to the present embodiment is possible to prepare line data in a reduced time because a gang of filtered image data is free of unnecessary image data (N, 64), (N, 63), (N, 126), and (N, 125), which are produced in the course of filtering the target image data subject to filtering.

[0481] The image processor according to the present embodiment initially sets up the number of pixels to be filtered. Such a simple setup provides subsequent edgewise pixel processing. This feature make it feasible to set up edgewise pixel-processing information at a smaller number of times.

[0482] The image processor according to the present embodiment includes the construction of that according to the first embodiment. Therefore, the image processor according to the present embodiment provides beneficial effects similar to those obtained in the first embodiment.

[0483] Having described preferred embodiments of the invention with reference to the accompanying drawings, it is to be understood that the invention is not limited to those precise embodiments, and that various changes and modifications may be effected therein by one skilled in the art without departing from the scope or spirit of the invention as defined in the appended claims.